# Computer-aided heuristic analysis of fluid models

KENNETH MAN-KAM YIP
Department of Computer Science
Yale University
P.O. Box 208285, Yale Station
New Haven, CT 06520-8285.
(Email: yip-ken@cs.yale.edu, Phone: 1-203-432-1213)

November 23, 1994

**Abstract**– A new kind of symbolic program to aid the heuristic simplification of fluid models is presented. The program, AOM, employs order of magnitude analysis and method of dominant balance to generate simplified models. It has two novel features: (1) it uses heuristic techniques to decide what equations to solve and what algebra to do, and (2) it explains its deduction steps. The basic operation of AOM consists of five steps: (1) assign order of magnitude estimates to terms in the equations, (2) find maximal terms of each equation, i.e., terms that are not dominated by any other terms in the same equation, (3) consider all possible n-term dominant balance assumptions, (4) propagate the effects of the balance assumptions, and (5) remove partial models based on inconsistent balance assumptions. AOM also exploits constraints among equations and submodels to simplify complicated fluid models such as the triple deck equations. Three annotated examples are presented to explain the operations of AOM. The implications for the development of computer-aided analysis programs for fluid dynamics and education are discussed.

## 1 Introduction

The use of computers as an integral part of fluid dynamics research is hardly new. "Computation" in the narrow sense of number-crunching has already made an important impact in fluid dynamics research and applications. It is likely that "computation" in the wide sense of symbolic (of which number is a special case) and geometric manipulation is going to make a similar if not bigger impact.

The purpose of this paper is to show it is possible to integrate the kind of qualitative and semi-intuitive reasoning fluid dynamics experts have with formal symbolic methods to simplify non-trivial flow models. The paper describes a new kind of symbolic program, AOM, that relies on heuristics and asymptotic order of magnitude analysis to simplify equations. Unlike earlier applications of computer algebra to fluid mechanics which concern the automatic extension of perturbation series to high order (Van Dyke (1984)), our work tackles the problem of how to select the appropriate time and space scalings in the first place so that the perturbation equations can be automatically derived.

There are three motivations for this work. The first motivation is to reduce substantially the time for simplifying fluid models. The program when combined with a conventional computer algebra system can automate the tedious perturbation calculations in simplifying a complicated set of equations. The fully automatic program can do what a user-guided computer algebra program can, but many times faster. An example of such application is the automatic determination of amplitude equations for chaotic wave motion in a wave tank. The physics of the problem is explained in Tsai *et al.* (1990), while the automated calculation is described in Yip (1992).

The second motivation is to raise the complexity limit of the kind of model approximation that can be performed today. For example, it is known that certain "universal" nonlinear partial differential equations – KdV and NLS, for example – can be obtained by rescalings and asymptotic expansions from a variety of nonlinear evolution problems (Ablowitz and Segur (1981)). Simplification of model equations involving multiple parameters and non-trivial geometry can in principle be done today, but they are just too complicated to do without more sophisticated computer help. Combining automatic scaling analysis with numerical techniques has also been shown to be effective in improving efficiency of numerical computation, and in determining space-time resolutions for certain CFD applications (Russo and Peskin (1990); Lomax and Mehta (1984))

The third motivation is to develop new computer representation and reasoning mechanisms to formalize expert reasoning in fluid dynamics. As educators, we are very interested in formalizing the informal expert knowledge so that the subject can be taught efficiently and cheaply to future scientists. AOM can explain its own reasoning steps and therefore can be part of a computer-aided tutoring system for teaching model simplification. We also see the potential for the development of a flexible problem solving environment for fluid dynamicists, where computer programs can "understand" more of human expert's problem solving tactics, and can function with less hand-holding from the users.

We define the *model simplification* problem as follows:

Given:

- a general flow model, such as the Navier-Stokes equations, or the Reynolds averaged equations,
- a set of boundary conditions, such as free-stream velocity, or the no-slip condition, and
- the order of magnitude estimates for a subset of the variables appearing in the model, such as the body extent in the flow direction, or the fluctuating velocity components.

Produce:

- one or more simplified models (i.e., models with many fewer terms) consistent with the problem specification,
- the set of constraints on flow parameters derived as a consequence of the simplifying assumptions (e.g., the boundary layer thickness $\delta = \frac{1}{\sqrt{Re}}$ for the flow over semi-infinite flat plate),
- a record of the reasoning steps leading to the approximate models.

The utility of simplified models cannot be overemphasized even when numerical solutions of a "complete" model are feasible. First, the numerical solutions may be too expensive. Second, the numerical results may be hard to interpret because the complete model might contain many details not essential for the particular flow physics in question. Third, good simplified models advance conceptual understanding.

Our approach is based on two ideas: self-consistent balancing of dominant terms, and asymptotic order of magnitude reasoning. The idea of dominant balance is described informally in Bender and Orszag (1978) for ordinary differential equations, and more formally in Echkaus (1979) in terms of "significant approximations." What is novel about AOM is its use of these heuristic ideas to guide its search for significant approximations, and its ability to explain its deduction steps.

Many boundary-layer type flows can be simplified by the following 7-step procedure (to be explained more fully in section 3):

```
1. Break up the region of interest into qualitatively distinct subregions
2. Estimate the order of magnitudes for a subset of quantities
3. Determine the form of gauge functions
4. Enumerate balancing possibilities
5. Deduce consequences of balancing assumptions on parameters
6. Match boundaries and solve simplified equations to get additional constraints on parameters.
7. Solve the resulting set of constraints
```

AOM automates the last five steps. The best way to understand AOM is to follow it out on some examples. We first illustrate the operations of AOM by three examples: Prandtl's boundary layer approximations for flow over flat plat, the far-field approximation of free turbulent wake, and the triple-deck approximations of laminar flow near the trailing edge of a flat plate. Even though all these examples are well-known, we believe AOM's analysis is more systematic and more understandable. Next we explain how AOM works. We then discuss the limitations of our approach. Finally we point out the possible implications of this work for building flexible computer-aided analysis programs for fluid dynamicists and for engineering education.

## 2 Three Examples

### Example 1

Let us first consider a simple example: the steady flow of an incompressible, viscous fluid over a flat plate (Figure 1). The following assumptions are usually made: the flow is 2D, steady, laminar, attached, and the thickness of the boundary layer region is much smaller than the length of the plate. The usual argument to derive the boundary layer equation goes like this. Since the streamwise velocity $u$ and distance $x$ are order unity, and transverse distance $y = O(\delta) \ll 1$, the continuity equation implies $v = O(\delta)$. The streamwise inertia terms are all order unity. On the hand, the viscous terms $\frac{\partial^2 u}{\partial x^2} \ll \frac{\partial^2 u}{\partial y^2}$ so that the first can be neglected. Balancing the inertia terms and the viscous term $\frac{\partial^2 u}{\partial y^2}$ in the streamwise equation reveals that $Re = O(\frac{1}{\delta^2})$. Both the inertia and viscous terms in the transverse equation at most $O(\delta)$. So the pressure term $\frac{\partial p}{\partial y} \approx 0$ to first approximation. Hence the transverse equation is reduced to the condition that the pressure is constant across the boundary layer.

We now show how AOM reproduces this line of reasoning. Assuming dimensionless Navier-Stokes equations, we encode the flow over flat plate as follows:

```
(flow-model  prandtl-boundary-layer
  (independent-variables
   (x (direction streamwise) (type space))
   (y (direction transverse) (type space)))
  (dependent-variables
   (U (depends-on x y) (direction streamwise) (type velocity))
```

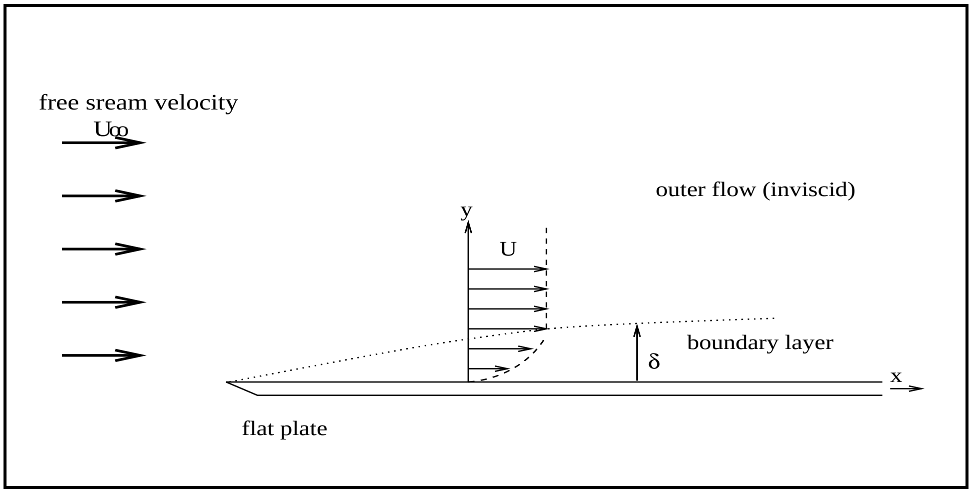


Figure 1: Boundary layer over a semi-infinite flat plate. The velocity gradient near the surface is large because of the no-slip condition. As one moves away from the surface, along the y-direction, the local velocity increases steadily as it approaches the free stream velocity. The thickness of the boundary layer is denoted by $\delta$.

```
 (V (depends-on x y) (direction transverse) (type velocity))
 (P (depends-on x y) (type pressure)))
(parameters
 (Re  delta))
(equations
 (streamwise-momentum-equation
  (U * (d U / d x) + V * (d U / d y)
     = - (d P / d x) + (d2 U / d2 x) / Re + (d2 U / d2 y) / Re))
 (transverse-momentum-equation
  (U * (d V / d x) + V * (d V / d y)
     = - (d P / d y) + (d2 V / d2 x) / Re + (d2 V / d2 y) / Re))
 (continuity
  ((d U / d x) + (d V / d y) = 0)))
(boundary-conditions
 (at (= y 0) (= U 0))
 (at (= y 0) (= V 0))
 (at (= y +inf) (= U 1)))
(essential-terms
 (viscous inertia))
(order-of-magnitudes
 (>> Re 1)
 (<< delta 1)
 (== U 1)
 (== x 1)
 (== y delta)))
```

Let me explain the notation. The flow model is given a name, `prandtl-boundary-layer`. Next the independent variables (`x` and `y`), dependent variables (U,V, and P), and parameters

(`Re` and `delta`) are declared. (For readability, we use a symbol (e.g., `delta`) and its greek letter (e.g., $\delta$) interchangeably.) For each variable, one assigns physical features (such as direction and type) and dependency information. For instance, U is the streamwise velocity, depending on both `x` and `y`. After the declarations of variables and parameters, the equations of motion and the boundary conditions are given. The momentum and continuity equations are given in infix notation for readability only; they are internally represented in prefix notation. The notation `(d U / d x)` stands for $\frac{\partial U}{\partial x}$, the partial derivative of U with respect to x. The symbol `d2` stands for the second partial derivative. The predicate `(at (= y 0) (= U 0))` specifies that U = 0 at y = 0. Now the final two items of the encoding. The `essential-terms` slot allows hints to be given to AOM. For instance, the expression `(essential-terms (viscous inertia))` instructs AOM that the simplified equations must contain at least one viscous term and one inertia term. Finally, AOM is given the order of magnitude of some parameters and variables. For example, `Re` is a large parameter greater than unity, while $\delta$ is a small parameter. The dimensionless U and `x` have order unity, while `y` = O($\delta$).

After the flow is encoded, we ask AOM to deduce the order of magnitude estimate for each term in the equations.

```
> (analyze prandtl-boundary-layer)
(V          δ)
(DV         δ)
(D2VDX2     δ)
(D2UDY2/RE  1/δ² × 1/Re)
(U          1)
(DU         1)
(DU-X       1)
(Y          δ)
(D2UDX2     1)
(DPDY       ?)
(DU-Y       1)
(DVDX       δ)
(P          ?)
(UDUDX      1)
(DUDY       1/δ)
(DP-Y       ?)
(VDVDY      δ)
(UDVDX      δ)
(VDUDY      1)
(D2UDY2     1/δ²)
(DUDX       1)
(DV-X       δ)
(DP         ?)
(DVDY       1)
(DPDX       ?)
(D2VDY2/RE  1/δ × 1/Re)
(D2VDX2/RE  δ × 1/Re)
(X          1)
```

```
(DP-X        ?)
(D2VDY2      1/δ)
(DV-Y        δ)
(D2UDX2/RE   1/Re)
```

In the above transcript, the user prompt is indicated by the symbol `>`; the computer output, tidied up for presentation, is given in upper case. The analysis proceeds as follows. First, for each variable, parameter, and term in the equations, an internal quantity is created. For instance, DUDX is the name of the term $\frac{\partial U}{\partial x}$ in the continuity equation. Intermediate quantities are also created to represent dependent variables. For example, since the velocity U is a function of two variables `x` and `y`, its total differential can be written as:

$$dU = \underbrace{\frac{\partial U}{\partial x}dx}_{\text{dU-x}} + \underbrace{\frac{\partial U}{\partial y}dy}_{\text{dU-y}}$$

Intermediate quantities like DU, DU-X, and DU-Y do not normally appear in the output; we show them for the sake of explaining AOM's deduction steps.

After all the quantities are created, AOM begins an order of magnitude analysis. For example, the transverse velocity `v` is found to be $O(\delta)$. It arrives at this conclusion by a sequence of local deductions. First, since U is order unity and its boundary value is 0, the total differential dU = O(1). Then, assuming O(dU-X) = O(dU-Y), AOM infers that both dU-X and dU-Y are O(1). The assumption that the differentials dU-X and dU-Y have the same order is usually left implicit in explanations of the boundary layer approximation. Next, DUDX (i.e., $\frac{\partial U}{\partial x}$) is set to order unity because dU-X and X are order unity. From the continuity equation, AOM infers that DVDY (i.e., $\frac{\partial V}{\partial y}$) is also order unity. We hope the nature of the local deduction performed by AOM is pretty clear by now. The quantities are related in a network of mathematical constraints so that the order of magnitude of any unknown can be deduced locally from knowns. We abbreviate the remaining deduction steps as follows:

```
DVDY = O(1) and Y = O(δ) ⇒ DV-Y = O(δ)
DV-Y = O(δ) and O(DV-X) = O(DV-Y) ⇒ DV-X = O(δ)
DV-X = O(δ) and DV-Y = O(δ) ⇒
DV = O(δ) DV = O(δ) and lower-bound(V) = 0 ⇒ V = O(δ)
```

The deduction process cannot determine magnitude information for P, DP, DP-X, and DP-Y. A more elaborate global deduction process, which we shall present shortly, is needed to ascertain their values.

AOM also keeps a record of its deduction steps. We can ask it to give the reason for the current assignment of V:

```
> (print-reason 'v)
V has value DELTA set by TOTAL-VARIATION-CONSTRAINT with DV = DELTA.
> (print-reason 'dv)
```

```
DV has value DELTA set by MAXIMUM-CONSTRAINT with DV-X = DELTA and
DV-Y = DELTA.
> (print-reason 'dv-y)
DV-Y has value DELTA set by MULTIPLIER-CONSTRAINT with
DVDY = 1 and Y = DELTA.
```

The explanations are given in the form of mathematical constraints (such as the MULTIPLIER-CONSTRAINT) relating the knowns and unknowns. For instance, DV-Y has value `delta` because it is equal to the product of DVDY and Y, and DVDY is currently assigned 1 and Y assigned `delta`. By chaining these reasons, AOM can present a "proof" of its deduction based on elementary assumptions and mathematical constraints.

After the analysis of order of magnitude, we ask AOM to simplify the equations:

```
> (simplify prandtl-boundary-layer)
Making <MODEL-2: PRANDTL-BOUNDARY-LAYER-1> from <MODEL-1: PRANDTL-BOUNDARY-LAYER>...
Assigning the leading order terms of TRANSVERSE-MOMENTUM-EQUATION to be:
 DPDY
Significant terms in STREAMWISE-MOMENTUM-EQUATION are:
  UDUDX  VDUDY  D2UDY2/RE
STREAMWISE-MOMENTUM-EQUATION: 2 candidate dominant sets:
  (D2UDY2/RE VDUDY UDUDX)
  (D2UDY2/RE DPDX VDUDY UDUDX)
Making <MODEL-3: PRANDTL-BOUNDARY-LAYER-1> from <MODEL-2: PRANDTL-BOUNDARY-LAYER-1>...
Balancing 2 terms:
D2UDY2/RE          and
VDUDY
in STREAMWISE-MOMENTUM-EQUATION with 1 parameter assumption: Re = 1/δ²
Assigning the leading order terms of STREAMWISE-MOMENTUM-EQUATION to be:
 D2UDY2/RE and VDUDY and UDUDX
<MODEL-3: PRANDTL-BOUNDARY-LAYER-1> is self-consistent.
Making <MODEL-4: PRANDTL-BOUNDARY-LAYER-2> from <MODEL-2: PRANDTL-BOUNDARY-LAYER-1>...
Balancing 3 terms:
D2UDY2/RE          and
DPDX               and
VDUDY
in STREAMWISE-MOMENTUM-EQUATION with 1 parameter assumption: Re = 1/δ²
Assigning the leading order terms of STREAMWISE-MOMENTUM-EQUATION to be:
 D2UDY2/RE and DPDX and VDUDY and UDUDX
<MODEL-4: PRANDTL-BOUNDARY-LAYER-2> is self-consistent.
Two simplified models are found.
MODEL-3
(0 = (- (D P / D Y)))
(((U * (D U / D X)) + (V * (D U / D Y))) = ((D2 U / D2 Y) / RE))
((D U / D X) + (D V / D Y) = 0)
MODEL-4
(0 = (- (D P / D Y)))
(((U * (D U / D X)) + (V * (D U / D Y))) = (- (D P / D X) + ((D2 U / D2 Y) / RE)))
((D U / D X) + (D V / D Y) = 0)
```

The simplification proceeds as follows. AOM creates intermediate models as simplifying assumptions are incrementally added. Model-2 is a copy of the prandtl-boundary-layer flow model. It then compares the terms in the two momentum equations:

```
> (print-terms (get-equation prandtl-boundary-layer 'streamwise-momentum-equation))
The terms of STREAMWISE-MOMENTUM-EQUATION are:
 UDUDX                  1
 VDUDY                  1
 DPDX                   ?
 D2UDX2/RE              1/Re
 D2UDY2/RE              1/δ^2 × 1/Re
> (print-terms (get-equation prandtl-boundary-layer 'transverse-momentum-equation))
The terms of TRANSVERSE-MOMENTUM-EQUATION are:
 UDVDX                  δ
 VDVDY                  δ
 DPDY                   ?
 D2VDX2/RE              δ × 1/Re
 D2VDY2/RE              1/δ × 1/Re
```

It notices every term in the transverse equation except the pressure term is much smaller than some term in the streamwise equation. Therefore, to leading order the pressure term DPDY, whose magnitude is still unknown but can be at most $O(\delta)$ or $O(\frac{1}{\delta \text{ Re}})$, is the only term that might contribute. So AOM assigns DPDY as the leading order term in the transverse equation.

Next AOM finds the significant terms in the streamwise equation, i.e., those terms that satisfy three conditions: (1) they are maximal, where a term is maximal if it is not smaller than any other term in the equation, (2) they are larger than some maximal term in the other momentum equation, and (3) their type is essential according to the `essential-terms` slot in the encoding of the flow model. In this instance, three terms are significant: UDUDX, VDUDY, and D2UDY2/RE. AOM keeps all three terms. Notice the term D2UDX2 is neglected because it is much smaller than D2UDY2/RE. The remaining pressure term DPDX may or may not be smaller than these significant terms. So AOM creates two possible candidate models, model-3 and model-4, and check if they can be self-consistent.

For model-3, AOM balances the three dominant terms UDUDX, VDUDY, and D2UDY2/RE and obtains the constraint $Re = O(\frac{1}{\delta^2})$. The constraint now propagates to the transverse equation and establishes that DPDY must be order $\delta$. Since no contradiction arises, AOM declares model-3 to be self-consistent.

Model-4 goes through a similar analysis. Four dominant terms UDUDX, VDUDY, D2UDY2/RE, and DPDX are balanced, and the same constraint on `Re` is obtained. Model-4 is also self-consistent.

After the simplification, one can ask AOM why certain term is not included in the simplified model:

```
> (why-not-chosen 'D2UDX2 model4)
THE TERM IS MUCH SMALLER THAN D2UDY2.
```

The analysis of the flat plate problem takes about 15 secs on a Sparc 10/51. [1]

**Example 2**

The second example concerns the simplification of the Reynolds averaged equations in the farfield of a free turbulent wake (Figure 2).

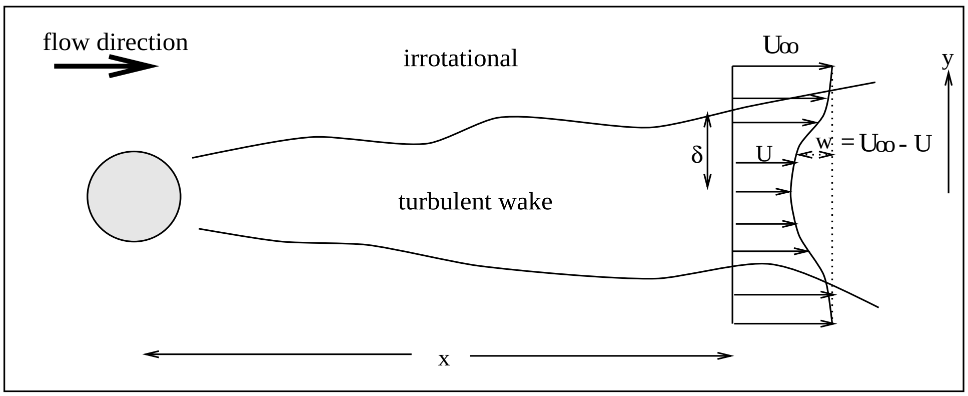


Figure 2: Free turbulent flow past solid body. Far from the body, the width of the wake grows as the square root of the distance downstream. The velocity profile for the mean flow shows a deficit within the wake. The velocity defect $w$ is defined as the difference between the free stream velocity and the mean flow velocity. The velocity profile recovers the free stream velocity $U_\infty$ in a distance of $O(\delta)$.

The new source of complexity is the addition of the fluctuating velocity terms and more parameters. A detailed description of the simplification process by order of magnitude arguments is given in Tennekes (1972), section 4.1. Let us see how AOM carries out a similar analysis. We first give it a flow model:

```
(flow-model turbulent-wake
  (independent-variables
   (x (direction streamwise) (type space))
   (y (direction transverse) (type space)))
  (dependent-variables
   (U (depends-on x y) (direction streamwise) (type velocity mean))
   (V (depends-on x y) (direction transverse) (type velocity mean))
   (w (depends-on x y) (direction streamwise) (type velocity))
   (uf (depends-on x y) (direction streamwise) (type velocity fluctuating))
   (vf (depends-on x y) (direction transverse) (type velocity fluctuating))
   (U-inf constant  (direction streamwise) (type velocity))
   (P (depends-on x y) (type pressure))
```

[1]The timing data in this paper are all based on a Sparc 10/51.

```
  (P0 constant (type pressure)))
 (parameters
  (Re delta u0 w0))
 (equations
  (streamwise-momentum-equation
   (U * (d U / d x) + V * (d U / d y)
      = - (d P / d x) + (d2 U / d2 x) / Re + (d2 U / d2 y) / Re
        - (d (uf * uf) / d x) - (d (uf * vf) / d y)))
  (transverse-momentum-equation
   (U * (d V / d x) + V * (d V / d y)
      = - (d P / d y) + (d2 V / d2 x) / Re + (d2 V / d2 y) / Re
        - (d (uf * vf) / d x) - (d (vf * vf) / d y))))
  (continuity
   ((d U / d x) + (d V / d y) = 0)))
 (boundary-conditions
  (at (= y ±inf) (= U U-inf))
  (at (= y ±inf) (= V 0))
  (at (= y ±inf) (= P P0)))
 (relations
  (U = U-inf - w))
 (essential-terms
  (turbulent inertia))
 (order-of-magnitudes
  (>> Re 1)
  (<< delta 1)
  (<< u0 1)
  (<< w0 1)
  (== U 1)
  (== U-inf 1)
  (== x 1)
  (== y delta)
  (== w w0)
  (== uf u0)
  (== vf u0)))
```

New terms are introduced in the encoding. The fluctuating velocity components are denoted by `uf` and `vf`. The velocity defect `w` is defined in the new `relation` slot. Scale parameters have the following meaning: $u_0$ characterizes the scale of the fluctuating velocities, $w_0$ the velocity defect, and $\delta$ the wake thickness. `U-inf` and `P0` are the free stream velocity and pressure; they are assumed to be constant. The hint given to AOM is that the simplified equations must contain some turbulent and some inertia terms.

Let us ask AOM to analyze this flow model:

```
> (analyze turbulent-wake)
> (print-terms (get-equation turbulent-wake 'transverse-momentum-equation))
The terms of TRANSVERSE-MOMENTUM-EQUATION are:
```
UDVDX $\quad w_0 \times \delta$

VDVDY $\quad w_0^2 \times \delta$

```
DPDY         ?
D2VDX2/RE    w0 δ / Re
D2VDY2/RE    w0 / (δ Re)
DUFVFDX      u0^2
DVF2DY       u0^2 / δ
Maximal sets are:
(DVF2DY), and
(D2VDY2/RE), and
(DPDY), and
(UDVDX).
Significant-Terms in TRANSVERSE-MOMENTUM-EQUATION = NIL
> (print-balancing-options (get-equation turbulent-wake 'transverse-momentum-equation))
TRANSVERSE-MOMENTUM-EQUATION: 6 candidate dominant sets:
(DVF2DY D2VDY2/RE)
(DVF2DY DPDY)
(DVF2DY UDVDX)
(D2VDY2/RE DPDY)
(D2VDY2/RE UDVDX)
(DPDY UDVDX)
```

AOM has done the order of magnitude analysis. (We skip all intermediate quantities in the printout.) It examines the transverse equation. Four maximal terms are found; none of them is significant though. Then we ask AOM to show the balancing options. It lists the six possible pairwise balances. If we wish, we could ask AOM to enumerate the 3-term and 4-term balances.

AOM then methodically works on each of the options. Each balancing introduces additional parameter constraints, which might lead to further order of magnitude analysis. For example,

```
> (simplify turbulent-wake)
Making <MODEL-2: TURBULENT-WAKE-1> from <MODEL-1: TURBULENT-WAKE>...
Balancing 2 terms:
DVF2DY          and
D2VDY2/RE
in TRANSVERSE-MOMENTUM-EQUATION
with 3 parameter assumptions:
(= RE (* (^ U0 -2) W0))
(>> U0 (* DELTA (^ W0 1/2)))
(<< RE (^ DELTA -2))
Assigning the leading order terms of TRANSVERSE-MOMENTUM-EQUATION to be:
 DVF2DY and D2VDY2/RE
Assigning the leading order terms of STREAMWISE-MOMENTUM-EQUATION to be:
 D2UDY2/RE
The model is not self-consistent because the simplified
equations do not contain the essential INERTIA term.
```

AOM works on the first of the 6 possible balances. Equating the order of magnitudes of the viscous and the turbulent term (D2VDY2/RE and DV2DY), AOM derives a new constraint: $Re = \frac{w_0}{u_0^2}$. [2] Under this balancing assumption, the remaining two terms DPDY and UDVDX

[2]We switch back to mathematical notation for readability.

are much smaller. So two additional constraints are obtained:

$$u_0 \gg \delta\sqrt{w_0} \tag{1}$$

$$Re \ll \frac{1}{\delta^2} \tag{2}$$

These new constraints invoke new order of magnitude analysis which allows AOM to conclude that the viscous term D2UDY2/RE in the streamwise equation dominates all the other terms.

| | |
|---|---|
| `> (print-terms (get-equation model-2 'streamwise-momentum-equation))` | |
| `The terms of STREAMWISE-MOMENTUM-EQUATION are:` | |
| UDUDX | $w_0$ |
| VDUDY | $w_0^2$ |
| DPDX | ? |
| D2UDX2/RE | $\frac{w_0}{Re}$ |
| D2UDY2/RE | $\frac{w_0}{\delta^2} \times \frac{1}{Re}$ |
| DUF2DX | $u_0^2$ |
| DUFVFDY | $\frac{u_0^2}{\delta}$ |
| `Maximal set is:` | |
| `(D2UDY2/RE)` | |

Since there are no inertia terms in the leading order approximation, AOM declares model-2 to be inconsistent.

Next, AOM repeats the above analysis for the next balancing options:

```
Making <MODEL-3: TURBULENT-WAKE-2> from <MODEL-1: TURBULENT-WAKE>...
Balancing 2 terms:
DVF2DY                    and
DPDY
in TRANSVERSE-MOMENTUM-EQUATION
with 2 parameter assumptions:
(>> RE (* (^ U0 -2) W0))
(>> U0 (* DELTA (^ W0 1/2)))
Assigning the leading order terms of TRANSVERSE-MOMENTUM-EQUATION to be:
 DVF2DY and DPDY
Adding <EQUATION: RELATION6>: (P0 = (P + (VF ^ 2)))
<MODEL-3: TURBULENT-WAKE-2> is self-consistent.
```

This time AOM gets two instead of three new constraints because the magnitude of the pressure term is unknown. The leading order of the transverse equation is:

$$\frac{\partial p}{\partial y} + \frac{\partial {v_f}^2}{\partial y} = 0$$

From this equation, AOM concludes the pressure outside the wake, P0, must be:

$$P0 = P + {v_f}^2$$

Propagation of the new constrains narrows down the maximal terms in the streamwise equation to three:

```
Maximal sets are:
(DUFVFDY), and
(D2UDY2/RE), and
(UDUDX).
Significant-Terms in STREAMWISE-MOMENTUM-EQUATION = NIL
STREAMWISE-MOMENTUM-EQUATION: 2 candidate dominant sets:
(DUFVFDY UDUDX)
(D2UDY2/RE UDUDX)
```

AOM reports only two candidate dominant sets because the third possible 2-term balance (DUFVFDY and D2UDY2/RE) does not contain any inertia terms. AOM then pursues the remaining two options:

```
Making <MODEL-4: TURBULENT-WAKE-1> from <MODEL-3: TURBULENT-WAKE-2>...
Balancing 2 terms:
DUFVFDY                    and
UDUDX
in STREAMWISE-MOMENTUM-EQUATION
with 3 parameter assumptions:
(= U0 (* (^ W0 1/2) (^ DELTA 1/2)))
(>> RE (* (^ DELTA -1) (* (^ U0 -2) W0)))
(>> RE (^ DELTA -2))
Assigning the leading order terms of STREAMWISE-MOMENTUM-EQUATION to be:
 DUFVFDY and UDUDX
<MODEL-4: TURBULENT-WAKE-1> is self-consistent.
Making <MODEL-5: TURBULENT-WAKE-2> from <MODEL-3: TURBULENT-WAKE-2>...
Balancing 2 terms:
D2UDY2/RE         and
UDUDX
in STREAMWISE-MOMENTUM-EQUATION
with 3 parameter assumptions:
(= RE (^ DELTA -2))
(<< RE (* (^ DELTA -1) (* (^ U0 -2) W0)))
(<< U0 (* (^ W0 1/2) (^ DELTA 1/2)))
Assigning the leading order terms of STREAMWISE-MOMENTUM-EQUATION to be:
 D2UDY2/RE and UDUDX
<MODEL-5: TURBULENT-WAKE-2> is self-consistent.
```

Both of these options turn out to be self-consistent. We ask AOM to print the simplified equations:

```
> (model-simplified-equations model-4)
(0 = (- (D P / D Y) + (- (D (VF ^ 2) / D Y))))
((U * (D U / D X)) = (- (D (UF * VF) / D Y)))
((D U / D X) + (D V / D Y) = 0)
> (model-simplified-equations model-5)
```

```
(0 = (- (D P / D Y) + (- (D (VF ^ 2) / D Y))))
((U * (D U / D X)) = ((D2 U / D2 Y) / RE))
((D U / D X) + (D V / D Y) = 0)
```

We can also ask AOM the magnitude of a flow variable say V, the vertical component of mean flow:

```
> (get-value 'V model-4)
(* W0 DELTA)
```

AOM continues to pursue the remaining four balancing options from model-2. It discovers none of them leads to a self-consistent simplification. The whole process for simplifying the turbulent wake problem takes about 25 secs.

### Example 3

In the last example, we show the kind of local deduction and global balancing heuristics of AOM can simplify a classic problem in laminar flow, flow over the trailing edge of a finite plate, and obtain the well-known triple-deck equations (Stewartson (1974)). The problem is interesting because the same structural argument seems to apply to a variety of other flow problems, including laminar separation, flow near a corner, a hump, a tilted trailing edge, and even supersonic flow.

The new source of complexity in this problem is the interaction among subregions in the triple deck model. Roughly the model has a 3-layer structure near the trailing edge (Figure 3): a "lower deck" reacts to the solid boundary just like Prandtl's thin boundary layer, an "upper deck" is controlled by the free stream condition, and a "main deck" transmits the displacement effect of the lower deck to the upper deck. Given this qualitative structural description, AOM's task to determine the size of each region and magnitude of the flow quantities such as velocity and pressure in each region.

```
(flow-model triple-deck
  (dependent-variables
   (U-inf constant (direction streamwise) (type velocity))
   (UB (depends-on y) (direction streamwise) (type velocity))
   (PO constant (type pressure)))
  (parameters
   (Re sigma-main sigma-upper pi-lower pi-upper pi-main alpha beta))
  (order-of-magnitudes
   (>> Re 1)
   (> alpha 0)
   (< alpha 1/2)
   (> beta 1/2))
  (regions
   (:layout vertical (lower-deck main-deck upper-deck))
   (lower-deck
     (inherit prandtl-boundary-layer)
     (dependent-variables
        (u1 (depends-on x y) (direction streamwise) (type velocity)))
     (relations
```

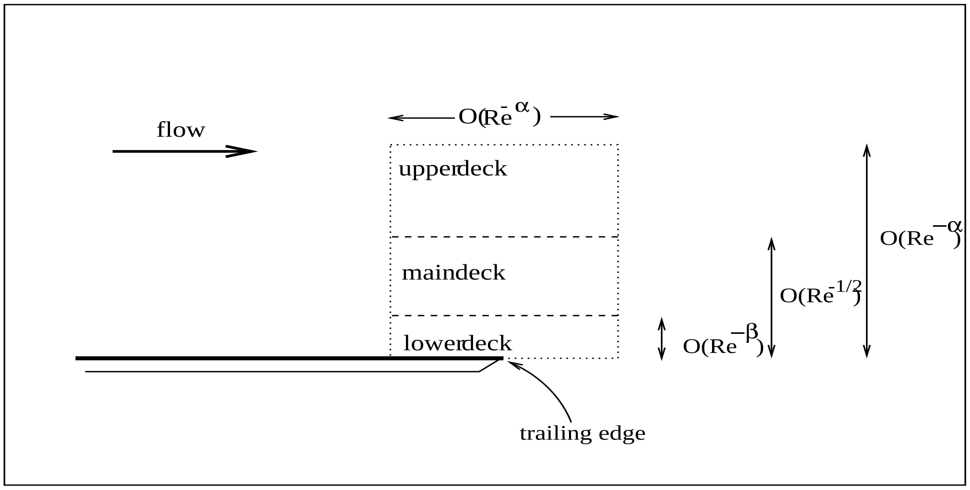


Figure 3: Schematic picture of the triple deck structure around the trailing edge of a finite plate. The three decks are marked by dotted lines. The horizontal dimension of the decks is $O(Re^{-\alpha})$. The vertical dimensions of the decks are as follows: $O(Re^{-\beta})$ for lower deck, $O(Re^{-\frac{1}{2}})$ for main deck, and $O(Re^{-\alpha})$ for upper deck. The qualitative three-deck structure is given to the program. The task of the program is to (1) deduce the values for unknown parameters such as $\alpha$ and $\beta$, and (2) simplify the flow equations in each deck.

```
      (U = (Re ^ (1/2 - beta)) * u1))
    (order-of-magnitudes
      (== u1 1)
      (== x (Re ^ (- alpha)))
      (== y (Re ^ (- beta)))
      (== P 'pi-lower)))
  (main-deck
    (inherit 2d-incompressible-steady-flow)
    (dependent-variables
      (u1 (depends-on x y) (direction streamwise) (type velocity)))
    (relations
      (U = UB + u1))
    (order-of-magnitudes
      (== u1 'sigma-main)
      (== x (Re ^ (- alpha)))
      (== y (Re ^ -1/2))
      (== P 'pi-main)))
  (upper-deck
    (inherit 2d-incompressible-steady-flow)
    (dependent-variables
      (u1 (depends-on x y) (direction streamwise) (type velocity)))
    (boundary-conditions
     (at (= y +inf) (= P P0)))
    (relations
      (U = U-inf + u1))
    (order-of-magnitudes
      (== u1 'sigma-upper)
      (== x (Re ^ (- alpha)))
      (== y (Re ^ (- alpha)))
```

```
(== P 'pi-upper)
(== U-inf 1)
(== P0 1)))))
```

Let us go over the encoding and explain the new features. This problem has eight parameters. `Re` is the usual Reynolds number. The three parameters, pi-lower, pi-main, and pi-upper, are the unknown magnitudes of the pressure in the respective decks. To make the expressions easier to read, we use the convention that hat ^ denotes the upper deck quantities, tilde ~ the main deck, and bar - the lower deck. For example, $\hat{\pi}$, $\tilde{\pi}$, and $\bar{\pi}$ represent the symbolic order of magnitudes for the pressure in the upper, main, and lower deck respectively. The two parameters, sigma-main ($\tilde{\sigma}$) and sigma-upper ($\hat{\sigma}$), are the unknown magnitudes of the streamwise velocity in the main and upper decks. Finally, alpha ($\alpha$) and beta ($\beta$) are unknowns characterizing the dimensionless size of the three decks. AOM task is two-fold: (1) find constraints governing these parameters, and (2) solve the constraints to determine the dependency of the seven unknowns on `Re`.

Two new features appear in the encoding. First, the `regions` slot allows the flow model to be made up of several submodels. In this case, the three submodels are named lower-deck, main-deck, and upper-deck. Declaration conventions within a submodel are the same as the parent model. Variables declared inside a submodel are considered local to that model. The `:layout` descriptor specifies that the subregions are stacked in the given order. Second, the `inherit` slot allows the momentum and continuity equations to be inherited from a previously defined flow model. For instance, the lower deck submodel inherits the Prandtl boundary layer equations (`model-4`) derived in example 1. We could have started with more general equations in which case AOM would repeat the analysis in example 1 to rederive the boundary layer equations. The main and upper decks just inherit the steady 2D incompressible Navier-Stokes equations.

Two types of hints are given in the encoding to AOM to solve this problem. First, the constraints on alpha and beta encodes the qualitative information that the lower deck is inside the Prandtl boundary layer (which is $O(Re^{-\frac{1}{2}})$), while the upper deck is outside. Second, the relations defining `u1` in the lower and main deck encode the assumption that the flows inside the decks are small perturbations from the Blausis profile, `UB(y)`. The size of the perturbations will come out of the analysis.

We next ask AOM to analyze the `triple-deck` problem. AOM first analyzes and simplifies each submodel in turn, going through the same process as described in examples 1 and 2. The result is summarized in Figure 4.

To derive the results, AOM finds constraints imposed on the parameters due to balancing and matching. Since there are seven unknowns, it needs to find seven equations for them. Let us see how this is done. The upper deck equations is reduced to the potential flow equations

Upper Deck:
UD-1

$$
\begin{aligned}
\frac{\partial u_1}{\partial x} &= -\frac{\partial p}{\partial x} \\
\frac{\partial v}{\partial x} &= -\frac{\partial p}{\partial y} \\
\frac{\partial u_1}{\partial x} + \frac{\partial v}{\partial y} &= 0
\end{aligned}
$$

Main Deck:
MD-1

$$
\begin{aligned}
UB\frac{\partial u_1}{\partial x} + v\frac{\partial UB}{\partial y} = 0 \\
\frac{\partial p}{\partial y} = 0 \\
\frac{\partial u_1}{\partial x} + \frac{\partial v}{\partial y} = 0
\end{aligned}
$$

MD-2

$$
\begin{aligned}
UB\frac{\partial u_1}{\partial x} + v\frac{\partial UB}{\partial y} &= -\frac{\partial p}{\partial x} \\
\frac{\partial p}{\partial y} &= 0 \\
\frac{\partial u_1}{\partial x} + \frac{\partial v}{\partial y} &= 0
\end{aligned}
$$

MD-3

$$
\begin{aligned}
UB\frac{\partial u_1}{\partial x} + v\frac{\partial UB}{\partial y} &= \frac{1}{Re}\frac{\partial^2 UB}{\partial y^2} \\
\frac{\partial p}{\partial y} &= 0 \\
\frac{\partial u_1}{\partial x} + \frac{\partial v}{\partial y} &= 0
\end{aligned}
$$

MD-4

$$
\begin{aligned}
UB\frac{\partial u_1}{\partial x} + v\frac{\partial UB}{\partial y} &= -\frac{\partial p}{\partial x} + \frac{1}{Re}\frac{\partial^2 UB}{\partial y^2} \\
\frac{\partial p}{\partial y} &= 0 \\
\frac{\partial u_1}{\partial x} + \frac{\partial v}{\partial y} &= 0
\end{aligned}
$$

Lower Deck:
LD-1:

$$
\begin{aligned}
u\frac{\partial u}{\partial x} + v\frac{\partial u}{\partial y} &= \frac{1}{Re}\frac{\partial^2 u}{\partial y^2} \\
\frac{\partial p}{\partial y} &= 0 \\
\frac{\partial u}{\partial x} + \frac{\partial v}{\partial y} &= 0
\end{aligned}
$$

LD-2:

$$
\begin{aligned}
u\frac{\partial u}{\partial x} + v\frac{\partial u}{\partial y} &= -\frac{\partial p}{\partial x} + \frac{1}{Re}\frac{\partial^2 u}{\partial y^2} \\
\frac{\partial p}{\partial y} &= 0 \\
\frac{\partial u}{\partial x} + \frac{\partial v}{\partial y} &= 0
\end{aligned}
$$

Figure 4: Results after calling `simplify-atomic-model` on each of the submodels of the triple deck problem. The simplified model for the upper deck is a potential flow. The main deck has four consistent models. The lower deck reduces to the Prandtl boundary layer equations. Eight composite models are possible. Only the combination {UD-1, MD-1, LD-2 } produces enough matching constraints for solving the seven unknowns.

(UD-1). Balancing the inertia term and the pressure term, AOM obtains one equation:

$$\hat{\pi} = \hat{\sigma}$$

The lower deck are the usual Prandtl boundary layer equations (LD-1 and LD-2). Balancing the inertia terms, pressure term, and the viscous term of the streamwise boundary layer equation produces two equations:

$$\begin{aligned}\bar{\pi} &= Re^{\beta-\frac{1}{2}-\alpha} \\ \alpha &= 3\beta - \frac{3}{2}\end{aligned}$$

The main deck has four possible simplifications: MD-1. MD-2, MD-3, and MD-4, and the lower deck has two. Therefore eight composite models are possible.

In the next phase of analysis, AOM extracts further consequences from the simplified equations. Two kinds of deductions are made. First, it checks if a simplified model belongs to a class of simple flows it knows about. For instance, it notices that the upper deck is a potential flow. So rules relating pressure and velocities for a potential flow are applicable. In this case, a rule asserts that the pressure on a thin body in a potential flow is proportional to the streamline slope. AOM therefore asserts $\hat{\pi} = O(\frac{v}{u}) = \hat{\sigma}$.

Second, simplified equations are checked if they can be further resolved to a simpler form. For example, the equations for MD-1 can be resolved to give:

$$\frac{\partial}{\partial y}(\frac{v}{UB}) = 0$$

Equations of the form:

$$\frac{\partial}{\partial y} Q = 0$$

are called **transmission constraints**, and Q is a transmission quantity. Physically, a transmission constraint asserts that the quantity Q does not vary in the y-direction across two regions. It implies a matching constraint on the quantity Q between two adjacent regions.

For each transmission constraint, the program asserts that the quantities corresponding to Q in two adjacent regions have the same order of magnitude. For example, the transmission constraint $\frac{\partial p}{\partial y}$ in MD-1 leads to two assertions:

$$\begin{aligned}\bar{\pi} &= \tilde{\pi} \\ \hat{\pi} &= \tilde{\pi}\end{aligned}$$

Similarly, the transmission constraint $\frac{\partial}{\partial y}\frac{v}{UB} = 0$ leads to two assertions:

$$\begin{aligned}\tilde{\sigma} Re^{\alpha-\frac{1}{2}} &= Re^{2\alpha-4\beta+\frac{3}{2}} \\ \tilde{\sigma} Re^{\alpha-\frac{1}{2}} &= \hat{\sigma}\end{aligned}$$

The final step is to solve all the parameter constraints that have been asserted. Seven such constraints are asserted for the triple deck problem. They are collected here for convenience:

$$
\begin{aligned}
\tilde{\sigma} Re^{\alpha-\frac{1}{2}} &= Re^{2\alpha-4\beta+\frac{3}{2}} \\
\tilde{\sigma} Re^{\alpha-\frac{1}{2}} &= \hat{\sigma} \\
\tilde{\pi} &= \bar{\pi} \\
\tilde{\pi} &= \hat{\pi} \\
\hat{\pi} &= \hat{\sigma} \\
\bar{\pi} &= Re^{\beta-\frac{1}{2}-\alpha} \\
\alpha &= 3\beta - \frac{3}{2}
\end{aligned}
$$

AOM has a built-in algebraic simplifier built on top of Mathematica. The simplifier augments Mathematica's algebraic capabilities: it handles inequalities and order of magnitude comparisons (see Yip (1993) for more details), and incorporates meta-rules controlling the simplification of algebraic expressions (Bundy (1983)). Our simplifier implements provably correct algorithms (Sacks (1987)). For the purpose of scaling analysis AOM does not require much algebraic help from Mathematica, thus avoiding the non-robustness of a typical computer algebra system (Stoutemyer (1991)).

The solutions for the parameters are given in Figure 5.

$$
\begin{aligned}
\alpha &= \frac{3}{8} \\
\beta &= \frac{5}{8} \\
\hat{\pi} = \tilde{\pi} = \bar{\pi} &= Re^{\frac{1}{4}} \\
\hat{\sigma} &= Re^{\frac{1}{4}} \\
\tilde{\sigma} &= Re^{\frac{1}{8}}
\end{aligned}
$$

Figure 5: Complete solution for the seven parameters in the triple deck problem.

Important physical information can be deduced from these parameters. The $\alpha$ and $\beta$ values establish the extent of the lower deck: the horizontal extent is $O(Re^{-\frac{3}{8}})$ and vertical $O(Re^{-\frac{5}{8}})$. The horizontal velocity $u$ in the lower deck is $O(Re^{(\frac{1}{2}-\beta)}) = O(Re^{-\frac{1}{8}})$ from which we calculate the velocity gradient $\frac{\partial u}{\partial y} = O(Re^{\frac{1}{2}})$. Using the velocity gradient and the definition of drag coefficient $C_D$ (Yih (1977)), and integrating over the horizontal extent of the lower deck, we arrive at the following correction to the drag due to the finite edge effect:

$$C_D = O(Re^{-\frac{7}{8}})$$

The conclusion that the finite edge effect contributes $O(Re^{-\frac{7}{8}})$ is non-trivial. Researchers had for years assumed that the next order correction to the drag on the flat plate is $O(Re^{-1})$

(Van Dyke (1975)). Notice that we arrive at the conclusion without explicitly solving the triple deck equations!

The analysis and simplification of example 3 takes about 60 secs.

# 3 How AOM Works

Given the encoding of a general flow model, AOM generates all possible simplified models consistent with the constraints specified in the model. AOM's capability depends on two essential ingredients: (1) equations are represented as a network of constraints among objects (i.e., data structures representing the parameters, terms of the equations, and other intermediate quantities) so that order of magnitude deduction can be made locally, and (2) use of global term balancing heuristics and matching conditions between submodels to eliminate inconsistent models.

**Local deductions**

Variables, parameters, and terms of the equations in the encoding are represented as objects with slots storing information about variable dependency, order of magnitude, and physical features (such as space, velocity, and pressure). The physical features are used to determine the meaning of derived quantities by simple rewrite rules. For instance, a velocity quantity differentiated by a space quantity gives a velocity-gradient quantity. The physical meaning of a term in the equation is determined in a similar fashion. For example, a term that is the product of a velocity quantity and a velocity gradient represents the convective inertia term.

Equations are represented as constraints on objects so that when all but one object are known the value of the remaining one can be computed in terms of the others. Our constraint language has seven primitives:

- The constant constraint, `(constant q v)`, asserts that $O(q) = v$.
- The maximum constraint, `(maximum q1 q2 q3)`, asserts that $O(q3) = \max(O(q1), O(q2))$. Example: `(maximum du-x du-y du)`.
- The multiplier constraint, `(multiplier q1 q2 q3)`, asserts that $O(q1) \times O(q2) = O(q3)$. Example: `(multiplier u dudx ududx)`.
- The exponential constraint, `(exponential q1 q2 q3)`, asserts that $O(q1^{q2}) = O(q3)$. Example: `(exponential x 2 x2)`.
- The equality constraint, `(== q1 q2)`, asserts that $O(q1) = O(q2)$. Example: the continuity equation (3) is represented by `(== dudx dvdy)`.
- The variation constraint, `(variation f x df-x)`, captures the inference that when the partial differential of a function $f(x, y)$ with respect to $x$ is much less than the value of $f$ at its outer boundary, then $f$ is asymptotically equal to its boundary value. Symbolically, $df\text{-}x = o(f_0) \Rightarrow O(f) = O(f_0)$, where $f_0$ is the value of $f$ at its outer boundary in the x-direction.

- The total-variation constraint, `(total-variation f df)`, asserts: $O(df) = O(\text{upperbound}(f) - \text{lowerbound}(f))$.

Most of these constraints are self-explanatory. In the variation constraint, it should be noted that the differentials *df* or *df-x* do not have to be infinitesimals. They represent the amount the function $f$ changes when the independent variables change. Even if *df* happens to be an infinitesimal, in AOM it is not true that *df* is less than any non-zero quantity. The reason is that asymptotic order of magnitude provides an ordering of infinitesimals (e.g., $\epsilon$, $\epsilon^2$, ...). So if $f_0 = O(\epsilon^2)$ and $df\text{-}x = O(\epsilon)$, then $df\text{-}x \neq o(f_0)$.

The constraint language allows simple inferences about quantities to be made. Consider the network generated by the continuity equation (Figure 6). Given that $u = O(1), x = O(1), y = O(\delta)$, the value $v = O(\delta)$ is automatically deduced, as described in example 1.

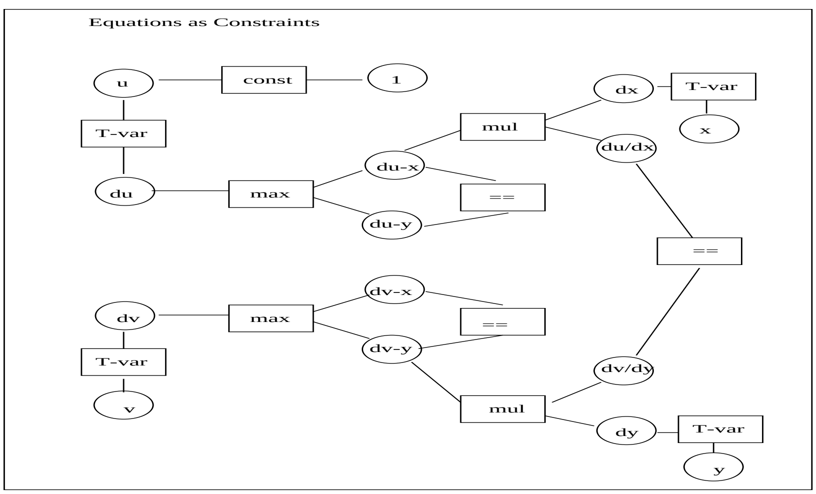


Figure 6: Equations are represented as a network of constraints for forward inferences.

Forward inferences result in the following estimates for quantities:

Streamwise equation:

$$\begin{aligned} u\frac{\partial u}{\partial x} &= O(1) \\ v\frac{\partial u}{\partial y} &= O(1) \\ \frac{\partial p}{\partial x} &= ? \\ \frac{1}{Re}\frac{\partial^2 u}{\partial x^2} &= O(\frac{1}{Re}) \\ \frac{1}{Re}\frac{\partial^2 u}{\partial y^2} &= O(\frac{1}{Re}\frac{1}{\delta^2}) \end{aligned}$$

Transverse equation:

$$\begin{aligned} u\frac{\partial v}{\partial x} &= O(\delta) \\ v\frac{\partial v}{\partial y} &= O(\delta) \\ \frac{\partial p}{\partial y} &= ? \\ \frac{1}{Re}\frac{\partial^2 v}{\partial x^2} &= O(\frac{1}{Re}\delta) \\ \frac{1}{Re}\frac{\partial^2 v}{\partial y^2} &= O(\frac{1}{Re}\frac{1}{\delta}) \end{aligned}$$

Continuity equation:

$$\begin{aligned} \frac{\partial u}{\partial x} &= O(1) \\ \frac{\partial v}{\partial y} &= O(1) \end{aligned}$$

### Asymptotic order of magnitude

To describe the order of magnitude of a quantity, we use the standard notation described in Van Dyke (1975). The description uses the order symbols $O$ ("big oh"), $o$ ("little oh"), and $\sim$ ("asymptotically equal") to express the relative magnitudes of two functions.

**Definition 1** $f(\epsilon) = O(g(\epsilon)), \epsilon \to 0$ if $\lim_{\epsilon \to 0} \mid \frac{f(\epsilon)}{g(\epsilon)} \mid = K$ where $K$ is a finite number.
**Definition 2** $f(\epsilon) = o(g(\epsilon)), \epsilon \to 0$ if $\lim_{\epsilon \to 0} \frac{f(\epsilon)}{g(\epsilon)} = 0$
**Definition 3** $f(\epsilon) \sim g(\epsilon), \epsilon \to 0$ if $\lim_{\epsilon \to 0} \mid \frac{f(\epsilon)}{g(\epsilon)} \mid = 1$

The notation $f \ll g$, which reads "$f$ is negligible compared to $g$", is synonymous with $f = o(g)$. The notation $f = O(g)$ implies the ratio of $\mid \frac{f}{g} \mid$ is bounded, whereas $f \sim g$ means the *relative error* between $f$ and $g$ goes to 0.

Below we list some useful rules of operation on order symbols implemented in AOM:

1. $O(fg) = O(f)O(g)$
2. $O(f+g) = \max(O(f), O(g))$
3. $O(f) + o(f) = O(f)$
4. $o(fg) = O(f)o(g) = o(f)o(g)$
5. If $f = O(g)$, then $\int_0^\epsilon f(t)dt = O(\int_0^\epsilon \mid g(t) \mid dt)$ as $\epsilon \to 0$.

Proofs of these relations can be found in Erdelyi (1956). Order relations cannot in general be differentiated. That is, if $f = O(g)$, then it is not generally true that $f' = O(g')$. However, using the definition of the total differential of a function $f(x,y)$, $df = \underbrace{\frac{\partial f}{\partial x}dx}_{df\text{-}x} + \underbrace{\frac{\partial f}{\partial y}dy}_{df\text{-}y}$ where $df\text{-}x$ and $df\text{-}y$ are the partial differentials, we can derive some useful rules involving partial derivatives:

1. $O(\frac{\partial f}{\partial x})O(dx) = O(df\text{-}x)$
2. $O(\frac{\partial f}{\partial y})O(dy) = O(df\text{-}y)$
3. $O(df) = \max(O(df\text{-}x), O(df\text{-}y))$

### Dominant balance heuristics

A powerful idea to simplify an equation is to identify the small terms in the equation, drop these terms, solve the simplified equation, and check for consistency (Bender and Orszag (1978); Lin and Segel (1974)). But this does not always work. Consider the following simple polynomial:

$$3\epsilon^2 x^3 + x^2 - \epsilon x - 4 = 0$$

in the limit $\epsilon \to 0$. We might naively drop the cubic and the linear terms because their coefficients are small. But if we do that, we only get two roots $x = \pm 2$, losing the third root. Thus, the process of simplification can lead to a loss of important information. We have a singular perturbation problem when the reduced equation changes order. Many interesting problems in fluid mechanics have this singular character (see Van Dyke (1975)).

What went wrong? The problem is that terms that appear small are not really small. The missing root depends inversely on $\epsilon$ in such a way that the cubic term is not negligible even its coefficient becomes small. To fix this problem, we introduce three concepts: an **undetermined gauge**, a **significant gauge**, and a **maximal set**. To begin, we will assume $x = O(\epsilon^n)$ where $n$ is still undetermined – hence the name undetermined gauge. The order of each term is then:

$$\underbrace{3\epsilon^2 x^3}_{O(\epsilon^{3n+2})} + \underbrace{x^2}_{O(\epsilon^{2n})} - \underbrace{\epsilon x}_{O(\epsilon^{n+1})} - \underbrace{4}_{O(1)} = 0$$

To determine the relative importance of terms, we use the heuristic that we only retain the smallest number of terms that will balance the equation. Since we must allow the situation where two or more terms may have the same asymptotic order, we group terms into equivalence classes by the relation $\sim$. A maximal set is any such class that is not smaller than any other classes. As an example, the cubic polynomial above has four maximal sets each containing one term. The heuristic can then be stated as follows:

**Heuristic of minimal complication** (or Method of Dominant Balance):
*If the equation has $n$ maximal sets, balance $m(\leq n)$ of them; these $m$ maximal sets are called dominant. Assume the remaining $(n-m)$ sets are negligible. Self-consistent choices of dominant maximal sets correspond to significant simplified equations.*

Our heuristic is a straightforward extension of the more common 2-term dominant balance procedure (Bender and Orszag (1978)): it allows multiple terms with the same order of magnitude, and it is applicable to PDEs. There are equations for which balancing two maximal sets does not give any self-consistent approximations. For instance, the ODE $\frac{dy}{dx} - \frac{y}{x} = \frac{\cos x}{x^2}$ requires a 3-term balance because all the pairwise balances are inconsistent. Values of $m$ larger than 3 are not that useful because first the resulting simplification will not be much different from the equation we start with, and second it rapidly increases the number of possible models to be tested.

To understand how the heuristic works, let's consider the cubic equation. Here the number of maximal sets $n$ is 4. All 1-set balances are inconsistent because they are singleton sets. Six possibilities are given by 2-set balances. For instance, one possibility is that the first two terms are dominant, i.e., $\epsilon^2 x^3 \sim x^2 \gg \epsilon x, 4$. Equating the two undetermined gauges, we get $3n + 2 = 2n$ and this implies $n = -2$. The remaining terms are $O(\epsilon^{-1})$ and $O(1)$, which is consistent with the assumption that the first two terms are dominant. So this possibility is included. On the other hand, if we assume $\epsilon^2 x^3 \sim \epsilon x \gg x^2, 4$, we get $n = -\frac{1}{2}$. But then $x^2 = O(\epsilon^{-1}) \gg O(\epsilon^{\frac{1}{2}})$, violating the assumption that it should be much smaller than the first term. This possibility must be excluded. A similar analysis shows that only one more possibility, when the second and fourth terms are dominant, i.e., $n = 0$, is self-consistent. So the heuristic concludes that we should consider *two* simplified polynomials:

$$3\epsilon^2 x^3 + x^2 = 0 \Rightarrow x \sim \frac{1}{3\epsilon^2}$$

and

$$x^2 - 4 = 0 \Rightarrow x \sim \pm 2$$

The values of $\epsilon^n$ for which we get self-consistent dominant maximal sets are called **significant gauges**. Notice that each significant gauge generates an equation that captures a qualitatively significant asymptotic behavior. As one can easily check, no 3-set balance is consistent for the cubic equation and therefore the two simplified polynomials are the only significant simplifications.

To illustrate the use of maximal sets, we consider the boundary layer equations. Scaling the independent variable $y$ by an unknown gauge $\delta \ll 1$ and applying the order of magnitude rules, we get the following estimates for the terms in the streamwise equation (see example 1):

$$u\frac{\partial u}{\partial x} = O(1);\ v\frac{\partial u}{\partial y} = O(1);\ \frac{\partial p}{\partial x} = ?;\ \frac{1}{Re}\frac{\partial^2 u}{\partial x^2} = O(\frac{1}{Re});\ \frac{1}{Re}\frac{\partial^2 u}{\partial y^2} = O(\frac{1}{Re}\frac{1}{\delta^2});$$

where the symbol '?' indicates that the estimate for a quantity is unknown. Since $\delta \ll 1$, we have $\frac{1}{Re} \ll \frac{1}{Re}\frac{1}{\delta^2}$. So the term $\frac{1}{Re}\frac{\partial^2 u}{\partial x^2}$ is not maximal. The remaining four terms are maximal and grouped into three sets:

$$A = \{u\frac{\partial u}{\partial x}, v\frac{\partial u}{\partial y}\},\quad B = \{\frac{\partial p}{\partial x}\},\quad C = \{\frac{1}{Re}\frac{\partial^2 u}{\partial y^2}\}$$

Similarly, the estimates for the transverse equation are:

$$u\frac{\partial v}{\partial x} = O(\delta);\ v\frac{\partial v}{\partial y} = O(\delta);\ \frac{\partial p}{\partial y} = ?;\ \frac{1}{Re}\frac{\partial^2 v}{\partial x^2} = O(\frac{\delta}{Re});\ \frac{1}{Re}\frac{\partial^2 v}{\partial y^2} = O(\frac{1}{Re}\frac{1}{\delta});$$

and the maximal sets are:

$$D = \{u\frac{\partial v}{\partial x}, v\frac{\partial v}{\partial y}\},\quad E = \{\frac{\partial p}{\partial y}\},\quad F = \{\frac{1}{Re}\frac{\partial^2 v}{\partial y^2}\}$$

### Simplification Algorithm

Given a detailed model, the simplification algorithm finds all self-consistent simplified models by the maximal set balancing procedure. A simplified model is **self-consistent** if (1) the terms neglected are consistent with the dominant balance assumptions, and (2) it contains essential terms specified in the input. The algorithm terminates when every equation has only one maximal set, and all the constraints among unknown parameters have been computed and resolved. The algorithm consists of two procedures: `simplify-composite-model` and `simplify-atomic-model`, which implement the interface procedures `analyze` and `simplify` that we saw in the three examples.

The principal steps of the procedure `simplify-composite-model` are:

```
1. Preprocess the detailed model specification
2. Call simplify-atomic-model for each submodel
3. Generate all possible composite models from the simplified submodels.
4. Compute consequences of each composite model
5. Propagate all the transmission constraints among submodels
6. Collect and solve the constraint set for all unknown parameters.
```

The procedure `simplify-atomic-model` operates on an atomic model, i.e, one that has no submodels. Its purpose is to simplify all the equations in the model. It has six steps:

```
1. Assign order of magnitude estimates to all the terms of the equations.
2. Find the maximal sets.
3. If every equation has one maximal set, then return the model and done.
4. Otherwise, pick one unsimplified equation and consider all m-set balances.
5. Propagate the effects of the balance assumptions to eliminate inconsistent balances.
   Each consistent balance generates a new partially simplified model.
6. Call simplify-atomic-atom recursively on all the partially simplified models.
```

# 4 Evaluation

*When does the simplification heuristic fail?*

The simplification algorithm described in this paper has three major limitations. The first one can be handled by relatively straightforward extensions to the current program. The second one needs new machinery to generate and manipulate approximation assumptions. The last one requires a completely new method of attack.

The first limitation has to do with the class of gauge functions used in the order of magnitude descriptions. The program currently recognizes only powers and negative powers of $\epsilon$. It is well-known that more general gauge functions, such as logarithms and exponentials of $\epsilon$, [3] are required for many problems: the slender body approximation, and higher order matching in Stokes flow, just to name a few. Including more general gauges would complicate the solution matching procedure (Van Dyke (1975)), a topic we plan to pursue.

The second limitation is the dependence of AOM on hints about the qualitative structure of the simplified models. For instance, AOM is told that there should be three decks in example

[3] These are the so-called Hardy's L-functions (Hardy (1924)).

3. AOM would be much more useful if it can nondeterministically try out some simple models, compare their predictions with experimental data, and use the discrepancy between prediction and data to guide the construction of more elaborate models. For instance, in calculating the viscous drag on the flat plate, it might first propose the potential flow model. Noticing the potential flow model gives zero drag, it conjectures the importance of viscosity and looks for regions where a viscous term becomes important. Near the trailing edge of the plate, it finds the horizontal length scale is no longer large compared to the O($Re^{-\frac{1}{2}}$) boundary layer, and therefore a new inner viscous layer may come into existence.

The third limitation is more serious. The simplification algorithm assumes that a few significant length scales characterizes the physics of a problem. An important class of physics problems, such as those in critical phenomena, essentially involves effects of many wavelength scales (Wilson (1979)). Similar phenomenon occurs during the energy cascade via eddies of many length scales in a turbulent flow (Yakhot and Orszag (1987)). Problems like these fall outside the framework of model simplification described in this paper.

*How good are the approximate models?*

There is no simple answer to this question. First, it is known that solutions to a self-consistent approximate model derived by dominant balances can be grossly inaccurate. A simple example is an ill-conditioned set of linear algebraic equations, in which a small change in the coefficients can lead to a large change in the solution vector.

Second, existence and uniqueness theorems for PDEs are rare. One often does not know whether the approximate model has a solution at all or whether the solution if exists will be unique. The strongest claim one can make seems to be this: An approximate model that is *not* self-consistent is certainly a poor approximation. In practice, an approximate model is validated by subjecting its predictions to experimental and numerical checks. In fact, there still exists no theorem which speaks to the validity and accuracy of Prandtl's boundary layer approximation, but ninety years of experimental results leave little doubt of its validity and its value.

## 5 Conclusions

This paper describes a computer program AOM that integrates the qualitative, heuristic reasoning fluid dynamics experts have with formal symbolic methods to simplify flow models. AOM is based on three ideas: asymptotic order of functions, the dominant balance procedure, and constraint formulation due to parameter assumptions and boundary matching. The paper introduces a number of technical advances:

- It automates asymptotic order of magnitude reasoning to extend the range of problems solvable by computer.
- It articulates and explains the dominant balance method so that even a beginning graduate student can use the technique to simplify partial differential equations with several parameters.
- It produces more systematic explanations of the scalings used in classical fluid models like the triple deck.
- It provides the scalings necessary for the automatic application of formal perturbation expansion procedures.

Human experts are extremely good at doing analytical theory. It is unlikely that automated systems like AOM can compete with them soon. However, we believe that the goal of giving concrete forms to powerful ideas that experts exploit in their problem solving process is extremely worthwhile for three reasons. First, once an idea is captured explicitly, it can be examined, communicated, and improved on. We believe explicit representation of knowledge and reasoning methods in fluid dynamics in a readable and executable medium can in the long run be beneficial to the progress of many areas of fluid dynamics.

Second, we see the potential for computer-aided analysis programs that accept less explicit instructions from users and go beyond algebraic calculations or numerical crunching. AOM can be thought of as a program that helps user decide what algebra to do and what equations to solve. We believe that in solving complicated problems the "meta-knowledge" in deciding what to do is as important as the ability to carry out the actual calculations.

Third, programs that mimic experts and explain their methods of solution have obvious educational implications. We can use programs like AOM as a basis for an intelligent tutor – for teaching students or apprentice engineers the skill of model simplification and the use of heuristic techniques. The student can ask the computer to carry out certain tasks as well as interrupt at any point during the computation to ask the computer to explain what it is doing and why.

## Acknowledgement

This paper was greatly improved by suggestions from Michael Beetz, R. Bhaskar, K. Sreenivasan, W.T. Tsai, Dick Yue, and two anonymous referees. The author thanks Harold Abelson, Drew McDermott, Gerald Sussman, and Jack Wisdom for continual encouragement. The research was funded in part by NSF NYI Award ECS-9357773 and CCR-9109567.

## References


Ablowitz, M.J. and Segur, H. 1981. *Solitons and the Inverse Spectral Transform.* SIAM.

Bender, Carl and Orszag, Steven 1978. *Advanced Mathematical Methods for Scientists and Engineers.* McGraw-Hill.

Bundy, Alan 1983. *The computer modelling of mathematical reasoning.* Academic Press.

Echkaus, Wiktor 1979. *Asymptotic anlaysis of Singular Perturbations.* North Holland.

Erdelyi, A. 1956. *Asymptotic Expansions.* Dover.

Hardy, G.H. 1924. *Orders of Infinity.* Cambridge Tracts, No. 12, 2nd edition.

Lin, C.C. and Segel, L.A. 1974. *Mathematics applied to deterministic problems in the natural sciences.* MacMillan.

Lomax, H. and Mehta, U.B. 1984. Some physical and numerical aspects of computing the effects of viscosity on fluid flow. In *Computational methods in viscous flows.* Pineridge Press.

Russo, M.F. and Peskin, R.L. 1990. Automatically identifying the asymptotic behavior of nonlinear signularly perturbed boundary value problems. *Journal of Automated Reasoning* 6.

Sacks, Elisha P. 1987. Hierarchical reasoning about inequalities. In *Proceedings AAAI-87.*

Stewartson, K 1974. Multistructured boundary layers on flat plates and relations bodies. *Advances in Applied Mechanics* 14.

Stoutemyer, David R. 1991. Crimes and misdemeanors in the computer algebra trade. *Notices of the American Mathematican Society* 38.

Tennekes, J.L. H abd Lumley 1972. *A first course in turbulence.* MIT Press.

Tsai, Wu-ting; Yue, Dick; and Yip, Kenneth 1990. Resonantly excited regular and chaotic motions in a rectangular wave tank. *Journal of Fluid Mechanics* 216.

Van Dyke, Milton 1975. *Perturbation Methods in Fluid Mechanics.* The Parabolic Press, annotated edition.

Van Dyke, Milton 1984. Computer-extended series. *Annual Review of Fluid Mechanics* 16.

Wilson, Kenneth G. 1979. Problems in physics with many scales of length. *Scientific American* 241.

Yakhot, V. and Orszag, S. 1987. Renormalization group analysis of turbulence i: basic theory. *Journal of Scientific Computing* 1.

Yih, Chia-shun 1977. *Fluid Mechanics.* West River Press.

Yip, Kenneth 1992. Computer aided heuristic and qualitative reasoning in fluid mechanics. In Houstis, E.N. and Rice, J.R., editors 1992, *Artificial Intelligence, Expert Systems, and Symbolic Computing*. Elsevier Science Publishers.

Yip, Kenneth 1993. Model simplifications in fluid mechanics. In *Proceedings of the 11th National Conference on Artificial Intelligence*. To appear in the Artificial Intelligence Journal (1995).